Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Review:

Data center network architecture in cloud computing: review, taxonomy, and open research issues*

Han QI†[1], Muhammad SHIRAZ[1], Jie-yao LIU[1], Abdullah GANI†[1], Zulkanain ABDUL RAHMAN[2], Torki A. ALTAMEEM[3]

([1]*Mobile Cloud Computing Research Lab, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia*)

([2]*Department of History, Faculty of Arts and Social Sciences, University of Malaya, Kuala Lumpur 50603, Malaysia*)

([3]*Department of Computer Science, Riyadh Community College, King Saud University, Riyadh 11533, Saudi Arabia*)

†E-mail: hanqi@siswa.um.edu.my; abdullah@um.edu.my

Received Jan. 9, 2014; Revision accepted May 10, 2014; Crosschecked Aug. 13, 2014

Abstract: The data center network (DCN), which is an important component of data centers, consists of a large number of hosted servers and switches connected with high speed communication links. A DCN enables the deployment of resources centralization and on-demand access of the information and services of data centers to users. In recent years, the scale of the DCN has constantly increased with the widespread use of cloud-based services and the unprecedented amount of data delivery in/between data centers, whereas the traditional DCN architecture lacks aggregate bandwidth, scalability, and cost effectiveness for coping with the increasing demands of tenants in accessing the services of cloud data centers. Therefore, the design of a novel DCN architecture with the features of scalability, low cost, robustness, and energy conservation is required. This paper reviews the recent research findings and technologies of DCN architectures to identify the issues in the existing DCN architectures for cloud computing. We develop a taxonomy for the classification of the current DCN architectures, and also qualitatively analyze the traditional and contemporary DCN architectures. Moreover, the DCN architectures are compared on the basis of the significant characteristics, such as bandwidth, fault tolerance, scalability, overhead, and deployment cost. Finally, we put forward open research issues in the deployment of scalable, low-cost, robust, and energy-efficient DCN architecture, for data centers in computational clouds.

Key words: Data center network, Cloud computing, Architecture, Network topology
doi:10.1631/jzus.C1400013 **Document code:** A **CLC number:** TP393

1 Introduction

Cloud computing is a network-based computing model that provides services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), on demand (Armbrust *et al.*, 2010). Since the advent of mobile cloud computing, data volume has been increased tremendously on the Internet. The International Data Corporation (IDC) announced that the size of big-data generated reached 1.8 ZB (1.8 trillion GB) in 2011, 2.7 ZB in 2012, and will reach 35.2 ZB in 2020 (Gantz and Reinsel, 2012). The deployed data management and processing mechanisms in the data center network (DCN), such as the Google File System (GFS) (Ghemawat *et al.*, 2003), Hadoop Distributed File System (HDFS) (Borthakur, 2007), BigTable (Chang *et al.*, 2008), Dryad (Isard *et al.*, 2007), and MapReduce (Dean and Ghemawat, 2008), are responsible for managing and processing the massive data. As the above mentioned systems and

* Project supported by the Malaysian Ministry of Higher Education under the University of Malaya High Impact Research Grant (No. UM.C/HIR/MOHE/FCSIT/03)

applications are data and communication intensive (a simple Web search request may need cooperation among more than 1000 servers), the information exchange among remote nodes and local servers to process computation is increasing rapidly. Thereby, a great data traffic flow stress is taken to the data center, and the limited inter-node communication bandwidth among servers is becoming a serious bottleneck for DCN. To support such cloud services and important applications (e.g., scientific computations, financial analysis, massive data processing and warehousing, and utility computing), Amazon, Google, Salesforce.com, and other corporations have established large data centers around the world (Buyya *et al.*, 2008).

The motivations for building such data centers are both economic and technical (Greenberg *et al.*, 2009). Reasonable cost and elastic utilization according to the business requirements are considered for information technology (IT) investment of enterprises in DCNs. A cloud computing provider offers a large pool of high performance computing and storage resources that are shared among the end users. Users subscribe to the cloud computing services and receive computing and storage resources allocated on demand from the pool. A number of enterprises still have concerns about the cloud computing service models; for example, the network part of the data center has not seen much commoditization and still uses enterprise-class networking equipment. The cost of using enterprise-class network equipment is large (upwards of $12 million per month for a 100 000-server data center) and is not suitable for accommodating Internet-scale services in data centers. To be profitable, these data centers make better use of some lower-cost network equipment to achieve high utilization with agile end-to-end network capacity assignment and un-fragmented server pools (Greenberg *et al.*, 2008a).

Cloud-oriented data centers (CDCs) offer a shared computing resource model with higher quality of service (QoS) at a lower total cost of ownership. The main difference between CDC and traditional data centers is 'virtualization', which allows for massive scalability, virtualized resources, and on-demand utility computing. Table 1 shows the comparison between traditional DCN and cloud-oriented DCN in features. The cloud-oriented DCN is simple to organize, operate, and is more scalable. In a traditional DCN, servers are fixed in the hardware and an additional budget (e.g., hardware and the installation and maintenance) is required for upgrading and scaling up to more applications and users. In cloud-oriented DCN, in contrast, multiple servers are already in place. The virtualization is used to provide only the resources that a specific user demands, which gives cloud-oriented DCN a great scalability. In other words, the cost of cloud-oriented DCN is lower when compared to the traditional DCN. Also, the traditional DCN lacks network bandwidth, scalability, and faces the cost of coping with the increasing demands of tenants in accessing the services of CDCs.

It is common for a CDC to contain hundreds to thousands of servers in an economy of scale (Beloglazov and Buyya, 2010). A fundamental question for the DCN is how to effectively interconnect the number of exponentially increasing servers with fault-tolerance, high availability, and significant aggregate bandwidth. The architectural design of DCN significantly affects its total performance. Therefore, the design of a novel DCN architecture with the char-

Table 1 A comparison of the traditional data center network (DCN) and cloud-oriented DCN

Feature	Traditional DCN	Cloud-oriented DCN
Ownership	Servers and software belong to users, and infrastructure belongs to the DCN provider	All equipment belongs to the DCN provider
Management tool	Multiple	Standardized
Application	Hosts a large number of relatively small/medium-sized applications which run on a dedicated hardware	Runs a smaller number of very large applications
Fault-tolerance or degradation	Limited tolerance or graceful degradation	Needs fault-tolerance or graceful degradation
Hardware environment	Mixed	Homogeneous
Workload	Complex workload for server installation	Simple workload for server installation

acteristics of scalability, low cost, robustness, and energy conservation is required.

In recent years, the scale of the DCN has constantly increased with the widespread use of cloud-based services and the unprecedented amount of data delivery in/between data centers. The traditional DCN architecture such as tree-based and Clos network, however, lacks aggregate bandwidth, scalability, and faces the cost of coping with the increasing demands of tenants in accessing the services of CDCs. Therefore, the design of a novel DCN architecture with the features of scalability, low cost, robustness, and energy conservation is required.

This paper reviews the recent research findings and technologies of DCN architectures to identify issues in the existing DCN architectures for cloud computing. The following are the contributions of the paper: (1) developing a taxonomy for the classification of the current DCN, (2) analyzing the traditional and contemporary DCN architectures, (3) comparing the DCN architectures according to the significant features including scale, bandwidth, fault-tolerance, scalability, overhead, and deployment cost, and (4) identifying open research issues in the deployment of scalable, low-cost, robust, and energy-efficient DCN architectures for data centers in computational clouds.

2 Background

Cloud computing is emerging as a viable service model, and therefore Everything as a Service (XaaS) (Rimal *et al.*, 2009) is viewed as a significant trend, such as Software as a Service (SaaS), Platform as a Service (PaaS), Hardware as a Service (HaaS), Infrastructure as a Service (IaaS), Network as a Service (NaaS), Monitoring as a Service (MaaS), Database as a Service (DBaaS), Communications as a Service (CaaS), and Human as a Service (HuaaS) (Fig. 1).

IaaS (Bhardwaj *et al.*, 2010) is the delivery of computer infrastructure as a service. Aside from the higher flexibility, a key benefit of IaaS is the latest technology and usage-based payment scheme. In IaaS, the provider offers virtual resources (VR), physical resources (PR), storage, load balancers, and LAN and/or a virtual private network (VPN) to users. Users are responsible for setting up the operating system, installing their own application software, and patching and maintaining the operating

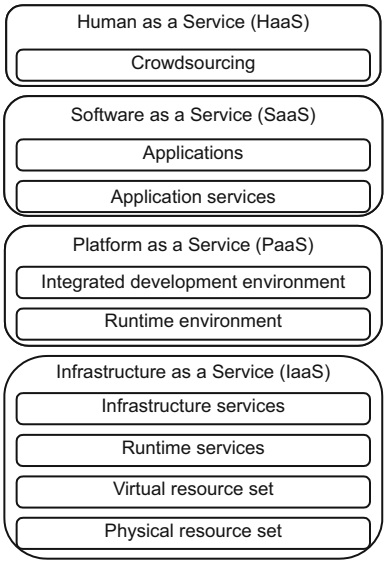

Fig. 1 Cloud layering concept

system and application software.

PaaS (Beimborn *et al.*, 2011) enables application developers with a platform, including all the systems and environments, to run their software solutions in a cloud-based environment without having to buy costly hardware. Compared with conventional application development, cloud providers offer a programming and execution environment, operating system, programming language, database, web server, and various available tools quickly. Key examples are Google App Engine (GAE) (Zahariev, 2009) and Microsoft Azure (Redkar and Guidici, 2011).

SaaS (Buxmann *et al.*, 2008) is a multi-tenant platform, commonly referred to as the application service provider (ASP) model, which offers application software, programming interfaces, elasticity, manages cloud infrastructure and platform, and charges typically on a monthly or yearly basis. Examples of the key providers are Microsoft Windows Live, Google Docs, and Salesforce.com.

HuaaS (Li and Svard, 2010) is an extension of XaaS to non-IT services. A group of humans can be used to perform tasks such as translation, design, research, and development. Key examples for HuaaS are Amazon Mechanical Turk, Microworkers, Wikipedia, and YouTube.

There are also many cloud computing systems besides Azure and GAE, like Amazon Web Service, IBM Smart Cloud (IBM® Smartcloud® Essentials, Packt Publishing Ltd.), and Nimbus (Shin *et al.*,

2012), which were created to support such services to users. Furthermore, Microsoft established a 707 000-square-foot DCN building in Chicago, 2009 (Vahdat *et al.*, 2010), which cost 500 million USD. There are 162 containers of 2500 servers, each with a total of 60 MW electricity in the building which cost 500 million US dollars. The Apple Data Center in Maiden was established in 2010 with 500 000 square feet and cost 1 billion dollars (Tarantino, 2012). Therefore, it is necessary to design a novel architecture to achieve high performance and high resource utilization using commodity hardware. The most effective technologies to achieve these were thought to be topology and switching.

As a basic hardware infrastructure of the data center and cloud computing, DCN has rapidly become a research focus. In recent years, top international conferences on computer science such as OSDI, ISCA, SIGCOMM, SIGMOD, and INFOCOM have proposed the topics relevant to DCN architecture. The leading international journals of the IEEE and ACM such as *IEEE Computing in Science and Engineering* and *IEEE/ACM Transactions on Networking* often publish DCN related papers. Universities and institutions such as Massachusetts Institute of Technology, Stanford University, the University of California Berkeley, Google, Amazon, Microsoft, and many others have established research groups focusing on DCN architecture.

The key goal of a new DCN architecture should be agility, elasticity, and as much throughput as possible, because the architectures have an impact on the overall properties of the DCN, such as how network devices connect to servers, how fast the switching can operate, how effective the routing protocol is applied in the system, and how complicated DCN deployment is. The following factors are motivation for investigating DCN architecture:

1. QoS in the upper layer: DCN architecture indicates the relationship of the positions of the servers in the data center, which relates to intermediate node links. The systems mentioned earlier, such as GFS (Ghemawat *et al.*, 2003) and HDFS (Borthakur, 2007), are achieved in the form of parallel and distributed computing through collaborative communication among a large number of servers in DCN. The quality of implementing these systems directly affects the QoS to the end user.

2. Deployment cost and energy consumption: A data center with different network architectures can accommodate different numbers of servers and switches. When the number of servers in DCN reaches tens of thousands or more, different network architectures result in huge data center deployment costs. For this reason, reducing the DCN deployment cost is seen by operators as a key driver for reaching a high cost/performance ratio and maximizing DCN profits. In addition, the power consumption of data centers has been pointed out as an amortized cost (Greenberg *et al.*, 2008a), which reaches 15% of the total cost of a data center. Moreover, a reliable power supply for a large-scale DCN needs a larger budget.

As major criteria, the metrics of scalability, cabling complexity, bandwidth, fault tolerance, and security should also be relevant to the novel DCN design, which will be discussed in detail in the following sections.

3 Data center network architectures

This section presents taxonomy for the classification of current DCN architectures and reviews the DCN architectures on the basis of taxonomy.

3.1 Taxonomy of data center network architectures

Current DCN architectures are classified into Clos/Tree, Valiant load balancing (VLB), hierarchical recursive architecture, and optical/wireless. Fig. 2 shows the taxonomy of current DCN architectures.

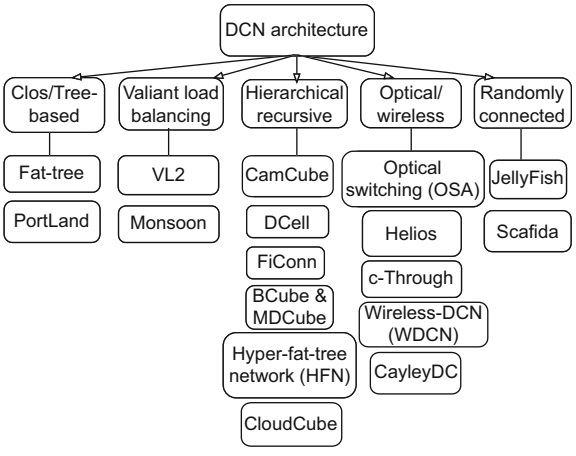

Fig. 2 Taxonomy of data center network architectures

The classification of the taxonomy is considered based on the features of current DCN architectures. Clos/Tree achieves high performance and high resource utilization using commodity hardware in a tree structure. VLB distributes traffic across a set of intermediate nodes and leverages the random distribution of traffic into multiple paths with equal cost. The hierarchical recursive architecture is proposed to avoid the existence of a single point of failure and to increase the network capacity. The optical/wireless architecture is established to include an optical and/or wireless network. The randomly connected is a random DCN where each switch or server can be randomly connected to some other switch.

In accordance with this taxonomy, each research trend with a selected research perspective is introduced in the following sections. Note that although each research relates to multiple perspectives listed in Fig. 2, we categorize them by selecting a key feature that shows the initial design motivations for each architecture.

3.2 Review on Clos/Tree DCN architectures

3.2.1 Tree-based architectures

The traditional data center network is a typical multi-root tree architecture, commonly composed of three layers of switches (three-tier) (Cisco Data Center, 2007). In the architecture, the top layer as a root is called the core layer, the middle layer is the aggregation layer, and the bottom layer is named the access layer. The higher layer devices possess a higher performance and value. The core layer typically is composed of several routers with redundancies accessing the external network on one side, implementing the external border gateway protocol (EBGP) or static routing protocol, and accessing the internal network on another side, implementing the interior gateway protocol (IGP). The accessing layer switches commonly provide 1 Gb/s and 10 Gb/s downlink and uplink interface, respectively. The aggregation layer switches normally have 10 Gb/s interfaces and allow aggregating between access layer switches and forwarding data. Fig. 3 gives a sample of the traditional tree-based hierarchical architecture.

In DCN, requests from the Internet are received by a core layer router and forwarded to the load balancing server in the aggregation layer. The load balancing servers maintain a mapping table that includes virtual IP address (VIP, for request acceptance) and direct IP address (DIP, for request processing). According to the table, the load balancing server forwards the Internet requests to the application pool in the accessing layer for processing.

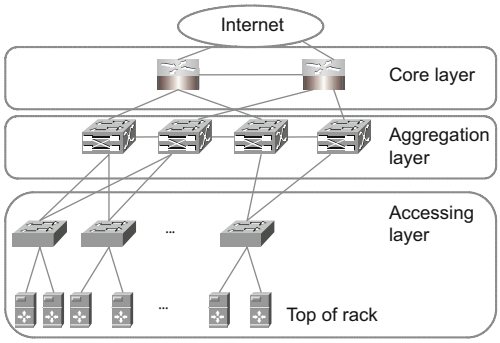

Fig. 3 A sample of the tree-based hierarchical data center network (DCN) architecture

There are many shortcomings in the traditional tree architecture (Cisco Data Center, 2007). First, the bandwidth increases significantly near the root of the tree, and deployment of a high performance network device is required, which may increase the cost. Second, the network scale is severely limited by the switch port. Third, the lower layer nodes will lose connection with others once the upper layer switch failure occurs. Last but not the least, with the increase in device processing capacity, there is not much doubt that data center power consumption will increase as well. Hence, researchers start to design alternative architectures for DCN.

3.2.2 Clos-based architectures

Clos is an enhanced architecture based on Tree, and is widely used in many enterprise-class data centers nowadays (Dally and Towles, 2004). The mathematical theory of Clos was introduced by Charles Clos from Bell Labs in 1953 for creating a non-blocking, multi-stage topology, which provides higher bandwidth than what a single switch is capable of supplying (Clos, 1953). A main feature of the architecture is multi-layer switching, wherein each switching unit connects to all units in the lower layer to reduce the number of intersecting nodes since input and output streaming is increasing. Fig. 4 shows an example of a three-stage folded Clos architecture.

In Clos, the leaf layer is responsible for advertising server subnets into the network fabric. The leaf layer determines oversubscription ratios, and thus

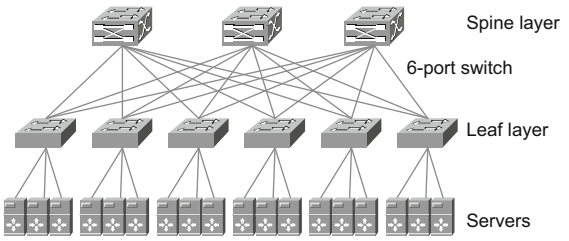

Fig. 4 Three-stage folded Clos topology

the size of the spine. The spine layer is responsible for interconnecting all leafs. As Clos is using a similar tree-based hierarchical data transmission mechanism, description is not necessary here. Though the multi-layer switching in Clos effectively reduces the stress of bandwidth restriction in the aggregation layer rather than the tree hierarchy, the same features and problems exist between the two architectures.

The above Tree and Clos architectures are initially designed for small or medium scale networks. In the era of cloud computing, however, a cloud-oriented data center is different from a traditional enterprise class data center as new requirements are desired for large-scale distributed computing since the number of data center network devices is growing rapidly. The main shortcomings of the tree-based hierarchical architecture include:

1. Bandwidth restriction: Data flow between aggregation layer servers is transferred through the core layer; however, the links between aggregation and core layers are normally over subscribed by a factor of 1:5 or more due to the equipment cost concern, which limits the communication among the servers in different branches of the tree, and leads to congestion and computation hot-spots even if the network capacity is available elsewhere (Greenberg *et al.*, 2009). In addition, the MapReduce application, for example, implements virtual machine (VM) migrating and deploys other bandwidth-intensive applications which increase the data traffic in DCN (almost to 80% of the whole traffic) and restrict the bandwidth availability between network devices.

2. Network scalability and reliability: In tree-based hierarchical architecture (Cisco Data Center, 2007), aggregation devices can support only up to 4000 servers due to the limited number of network ports and the requirement of a fast failure recovery mechanism. Therefore, it is difficult for the architectures to support the large number of servers in the data center for cloud computing. Moreover, due to the weakness of reliability in this architecture, if aggregation or core layer server failure occurs, the whole network performance is reduced (Bilal *et al.*, 2013b; Manzano *et al.*, 2013).

3. Resource fragmentation: Restricted bandwidth limits the performance of the data center as idle resources cannot be effectively assigned to the place where they are needed. The large spare resources capacity is often reserved by individual services or specific devices without sharing, for quick response to nearby servers once a network failure or demand request occurs. Moreover, the existing network scale in tree-based hierarchical DCN is IP addresses assigning and dividing servers by VLANs. Such IP address fragmentation limits the utility of VM migration among servers (the IP address has to be reconfigured with VM) and may bring a huge human configuration workload for the re-assignment.

4. Cost: Once the oversubscription-ratio changes occur in the aggregation and core layers, the only way to enhance the network performance is to upgrade high capacity devices. However, due to a larger price difference between highly advanced devices and commonly employed switches and routers, the upgrading cost can be very high. In addition, a mechanism of 1:1 equipment redundancy on the switches in upper layers is deployed in the tree-based hierarchical architecture to ensure the performance of DCN in situations of switch failure.

3.2.3 Fat-tree

To deal with the network bottleneck and upper layer single node failure, Al-Fares *et al.* (2008) introduced a Clos-based DCN architecture called fat-tree.

Similar to tree architecture, the switches in fat-tree are categorized into three layers: core layer, aggregation layer, and edge layer. Fig. 5 shows a classical fat-tree architecture. In this architecture, a range of switches in a square are called pod. In this diagram, there are $k = 4$ switches in each pod; half of them belong to edge switches and half are aggregation switches. Similarly, the aggregation switch uses each of $k/2$ ports while connecting to edge and core switches. Therefore, the maximum number of servers in fat-tree is $K^3/4$, and the maximum number of switches is $5K^2/4$.

Fat-tree uses the 10.0.0.0/8 private range to set the interior DCN address, and the format for the

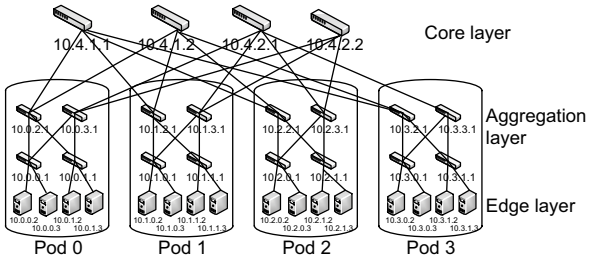

Fig. 5 A sample topology of fat-tree architecture (Bilal *et al.*, 2012)

pod switch is 10.pod.switch.1. The 'pod' indicates a pod number ([0, K–1]) and 'switch' means the position of the switch in the pod ([0, k–1], from left to right and bottom to top). The IP format for the core switch is 10.k.j.i, where j and i show the coordinates of switches between core switches and aggregation switches (start from top-left). The host IP format is described as 10.pod.switch.id, where 'id' means the host position in its own subnet.

Fat-tree improves the cost-effectiveness by deploying a large number of low-cost switches with complex connections to replace the expensive and more advanced switches in DCN. The equal number of links in different layers enables non-blocking communication among servers, which relieves the network bandwidth bottleneck. However, the scale of fat-tree architecture is restricted by the number of device ports. For example, a range of 48-port switches support only a maximum of 27 648 servers. Greenberg *et al.* (2008b) pointed out that the fat-tree architecture is very sensitive to low-layer switch failure and will impact the forwarding performance of DCN as it is still a tree-based structure. Facebook has used an architecture called the 'four-post network', which is composed of several layers of switches. This design was developed in response to previous network failures, as one switch going down would lead to a service outage.

3.2.4 ElasticTree

Due to the uncertainty of data traffic in DCN, Heller *et al.* (2010) pointed out that providing full bandwidth connection among all edge switches is not necessary. Hence, ElasticTree architecture is proposed from the perspective of power saving based on the fat-tree architecture. Turning on or off switches and connections on demand is the main feature of ElasticTree.

ElasticTree consists of three logical modules: optimizer, routing, and power control. As shown in Fig. 6, the optimizer aims to find the minimum power network subnet that satisfies current data flow conditions. Its inputs are network topology, the data flow matrix, the power model for each switch, and the desired fault-tolerance property. The optimizer outputs a set of active components to power control and routing modules. The power control module toggles the power states of ports, adapters, and entire switches. The routing module provides a route to the data flow.

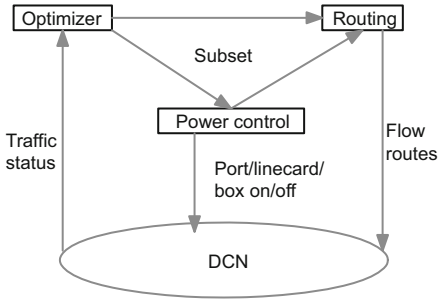

Fig. 6 A sample of ElasticTree network architecture

ElasticTree is designed for power saving in DCN, which effectively reduces the maintenance cost for the data center. However, as ElasticTree is deployed based on fat-tree, it has the same problem. Fig. 7 shows a sample topology of ElasticTree. Tschudi *et al.* (2004), Beloglazov and Buyya (2010), and Tzirita *et al.* (2013) have discussed in detail the energy efficiency within the data center in general and specifically in DCN.

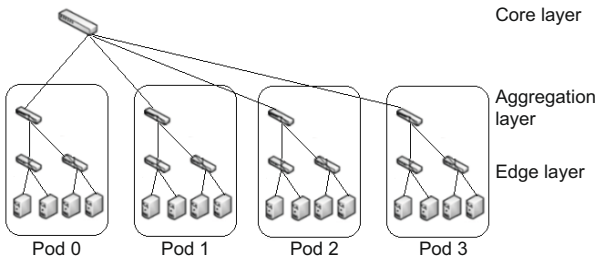

Fig. 7 A sample topology of ElasticTree DCN architecture

3.2.5 PortLand

Based on the fat-tree architecture, Niranjan Mysore *et al.* (2009) proposed a scalable and fault-tolerant two-layer routing fabric, PortLand, which employs a fabric manager and pseudo MAC address

(PMAC) to the forwarding data packet, and MAC to PMAC mapping to avoid modification in servers.

PortLand edge switches learn a unique pod number and a unique position number in each pod. A location discovery protocol is employed to assign these values. For all directly connected hosts, edge switches assign a 48-bit PMAC. The format of the PMAC is pod.position.port.vmid, where 'pod' (16 bits) indicates the pod number of the edge switch, 'position' (8 bits) reflects the switch position in the pod, and 'port' (8 bits) and 'vmid' (16 bits) describe the number of ports that the host connects to and the number of VMs deployed on the same physical machine (PM), respectively.

Whenever a source host desires to communicate with another host, it searches the target PMAC through the fabric manager. Once data packets reach the destination node, the ingress switch modifies the PMAC to actual MAC (AMAC) of the target. Upon completing VM migration from one PM to another, the fabric manager maintains the new PMAC to AMAC mapping and broadcasts to the previous PM where VMs were located before.

PortLand deploys a new two-layer based routing mechanism based on the fat-tree architecture, which supports a better fault-tolerant routing and forwarding, VM migration, and network scalability. However, modification of the existing switches is required to achieve the above features. In addition, as the fabric manager plays a major role in PortLand, the risk of single node failure still exists in this architecture.

3.3 Valiant load balancing DCN architecture

The Valiant load balancing (VLB) architecture was initially introduced by Valiant (1990) for processor interconnection networks, which is approved with capacity for handling traffic variation. VLB can achieve a hotspot free fabric for DCN when random traffic is divided into multiple paths.

3.3.1 VL2

VL2 is another tree-based architecture introduced by Greenberg *et al.* (2009) for dynamical resource allocation in DCN. The difference with fat-tree is that VL2 connects all servers through a virtual two-layer Ethernet, which is located in the same LAN as the servers. In this case, all servers can be assigned to upper layer applications as no resource fragmentation occurs (Fig. 8). VL2 uses the Clos topology to increase connection, and the VLB mechanism to assign routing for load balancing. Moreover, VL2 implements equal-cost multi-path (ECMP) routing to forward data over multiple optimal paths and resolve the problem of address redistribution in VM migration. Therefore, VL2 is considered in the VLB category.

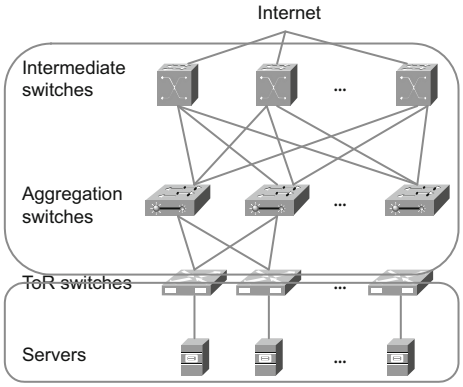

Fig. 8 A sample topology of VL2 DCN architecture

Since VL2 follows the traditional tree architecture in connection, it has been widely used to enhance the existing DCN. However, its network reliability is not improved and still has problems in scalability and single node failure.

3.3.2 Monsoon

The architecture of Monsoon (Greenberg *et al.*, 2008b) is as shown in Fig. 9, where over 100 000 servers are linked in a two-layer network without over subscription. The core border router and accessing router in layer 3 use ECMP for multi-path transmission, and VLB mechanism for load balancing like VL2.

Monsoon uses a MAC-in-MAC technology to create MAC layer tunnel, modifies the traditional address resolution protocol (ARP) to a user mode process, and allows a new MAC interface to forward encrypted Ethernet frames. These mechanisms and solutions, however, are not compatible with the existing Ethernet architecture. Bilal *et al.* (2013a) implemented and discussed in detail some DCNs from the perspective of various network loads and traffic.

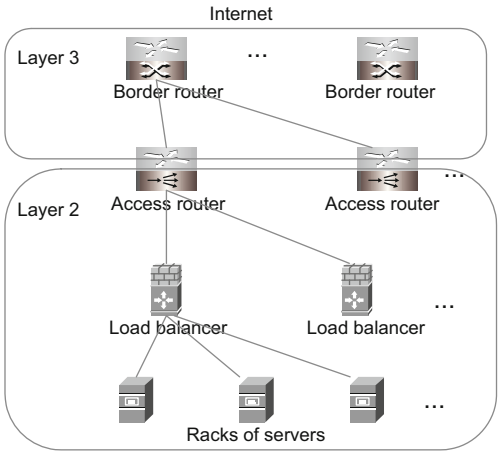

Fig. 9 A sample topology of Monsoon DCN architecture

3.4 Hierarchical recursive DCN architectures

Hierarchical recursive architecture is generally appropriate to avoid the bottleneck of single point failure and increase network capacity.

3.4.1 CamCube

CamCube, a non-switch architecture presented by Abu-Libdeh *et al.* (2010), constructs a network with a 3D torus topology directly by each server and connects with two neighbor servers in 3D directions. The topology of CamCube is as shown in Fig. 10.

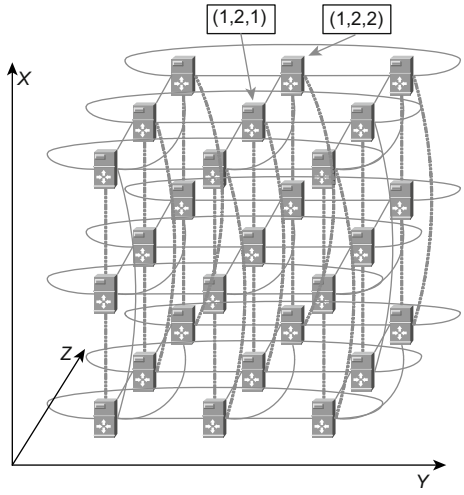

Fig. 10 A sample topology of CamCube architecture

CamCube assigns an(x, y, z) coordinate to indicate the position of each server in the topology, and provides functionality to send packets to or receive packets from one-hop neighbors. CamCube provides a platform for developers to create a more efficient routing algorithm for API according to the requirement, which decreases the additional network performance overhead and verifies the efficiency of this design.

CamCube has a simple structure and connection, and a high link redundancy. It is not a tree-based structure and thus there is no bandwidth bottleneck in a specific node. However, the servers play the role of a switch to forward data, which consumes part of the servers' computing resources and reduces the computing efficiency of servers. In addition, the number of network adapters installed in each server is limited (commonly two adapters for each server), which means the size of the CamCube network is also limited.

As CamCube has a relatively long routing path in a torus ($O \cdot N^{1/3}$ hops, with O denoting the maximum number of network flows and N the number of servers), which causes decrease in performance and increase in cost of DCN, Popa *et al.* (2010) introduced a De Bruijn-based DCN architecture where servers within a rack are labeled and connected as a De Bruijn graph structure. Those servers with the same label but in different racks are also connected as a De Bruijn structure. The diameter of the De Bruijn structure is $\log N$. This means that it has better routing performance and lower cost, in contrast with the CamCube structure.

The approach of using recursive structure DCN architectures, which relieves the bottleneck in core layer routers and provides multiple paths in pairs of servers, has been widely achieved in DCell, FiConn, BCube, MDCube, and HFN (Guo *et al.*, 2008; Ding *et al.*, 2012).

3.4.2 DCell

DCell (Guo *et al.*, 2008) is a recursively defined network architecture. As shown in Fig. 11, DCell_0 is a basic unit to construct a larger DCell which consists of n servers and a mini-switch. If there are m servers in the DCell_k network, DCell_{k+1} is considered as a compound graph structure consisting of $m+1$ DCell_k.

DCell uses a distributed routing algorithm called DCellRouting for data forwarding. According to the destination node and the relationship between server and virtual nodes, the data packet is forwarded to the next hop automatically without routing table search in the server. The massive link

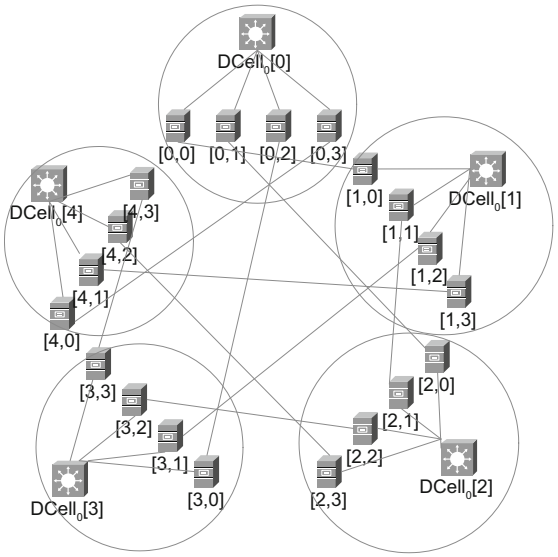

Fig. 11 A sample topology of DCell architecture

redundancy in DCell makes for a higher bandwidth than in a tree-based structure and has better performance in a one-to-all or all-to-all communication model in data-intensive computing. The situation of the server, link, and rack failure has been considered in DCell design. A data packet can also be delivered to a destination node through a fault-tolerant path when a failure is detected by the server or switch. In addition, DCell uses local reroute, local link-state, and a jump-up mechanism to address the above failures. As the routing algorithm in the DCell network is running between layers 2 and 3, the exciting TCP/IP protocol based applications can be deployed seamlessly and effectively in the structure.

One of the shortcomings of DCell architecture is that more interfaces and ports are needed to extend the network size. Furthermore, the lower level servers undertake more forwarding tasks and this load balancing will be a challenging issue to deal with in the future. Nevertheless, the proposed DCell architecture indicates a novel thinking, and has been a milestone in DCN research.

3.4.3 FiConn

A common commercial server typically has two network adapters, one for data receiving and forwarding and the other for redundancy. To reduce the additional overhead caused by massive link redundancy, Li *et al.* (2009) introduced a modified DCell structure, called FiConn. Similar to DCell, FiConn uses a compound graph creating its FiConn structure. In a four-level FiConn with 16-port switches, the number of servers can reach 3 553 776. Fig. 12 shows the one-level FiConn architecture.

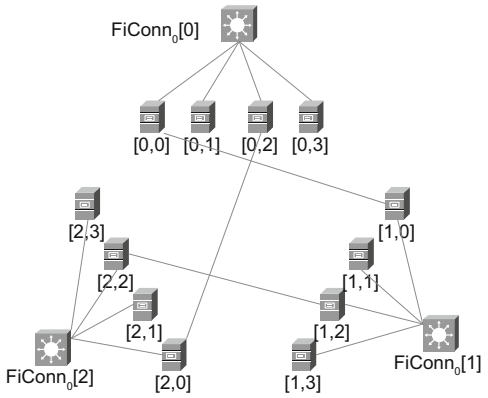

Fig. 12 A sample topology of one-level FiConn architecture

FiConn decreases the overhead when establishing the network by decreasing some performance. In contrast with DCell, the full connections among virtual nodes at the same level are not required in FiConn; instead, it uses only idle ports of servers and switches to connect with other devices, which decreases the number of redundant links and network adapters on the server. Therefore, we do not need to install multiple network adapters in the server and the number of ports required for the higher level switch is reduced, which means the cost of DCN establishment is decreased.

3.4.4 BCube and MDCube

Guo *et al.* (2009) proposed a hypercube related structure of a data center network, named BCube (Fig. 13). Similar to the recursively defined characteristic of DCell structure, BCube_0 is constructed by n servers connecting to an n-port switch, and BCube_1 is constructed from n BCube_0 connecting to n switches. More generically, a BCube_k is constructed from n BCube_{k-1} connecting to n^k n-port switches. Each host has $k+1$ parallel paths with different lengths. Thus, a k-level BCube structure, BCube_k, has n^{k+1} servers and $n^k(k+1)$ mini-switches. Each host has $k+1$ parallel paths in BCube but the lengths are different. BCube also makes a one-to-X speedup for data replication, and such a speedup depends on the number of network adapters.

One of the design goals of BCube is to establish a shipping-container based modular data cen-

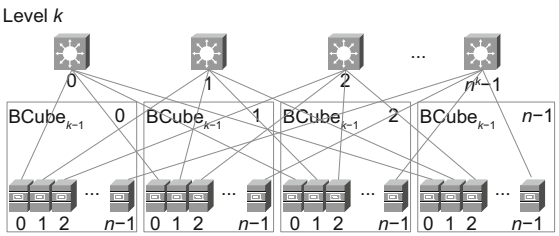

Fig. 13 A sample topology of BCube architecture

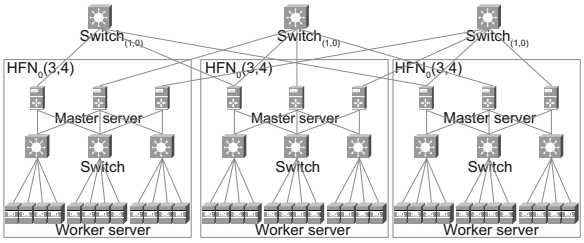

Fig. 14 A sample topology of HFN architecture with n=3, m=4

ter (MDC); in contrast, connecting these data centers and creating a larger data center are the main goals of MDCube (Wu *et al.*, 2009). MDCube is proposed as an interconnection structure among shipping-containers to construct by fiber a larger size of DCN. In MDCube, each BCube container is assumed as a virtual node, connecting with other nodes to create a HyperCube network structure.

Servers in BCube have multiple ports to support selectable routing, high fault-tolerance, and high throughput. Therefore, BCube has better performance in one-to-many and many-to-many communication, and resolves load balancing issues in lower level servers. However, the number of switches in BCube_k is k times that of DCell_k for connecting a certain number of servers, and thereby BCube is more costly in cabling layout and deployment than DCell.

3.4.5 Hyper-fat-tree network (HFN)

Optimal DCN for some specific requirements desired in cloud computing, such as HFN (Ding *et al.*, 2012) and CloudCube (Jericho Forum, 2009), is proposed for MapReduce optimization.

$\text{HFN}_{0(N,M)}$ is the basic building block of the entire network topology, which consists of n master servers, $n \times m$ worker servers, and n 'm-port' switches. Each switch connects m worker servers, n master servers, and n 'm-port' switches to create two-vertex sets of the bipartite graph. More generically, the level $k+1$ HFN, HFN_{k+1}, consists of n HFN_k and $n \times (k+1)$ 'n-port' switches. If all the HFN_0 are considered as virtual servers, it is obvious that the basic architecture of HFN is from BCube. A sample topology of HFN is as shown in Fig. 14.

In HFN, master servers control the entire procedure of MapReduce and receive tenant's requests. The server assigns a task to multiple master servers and forwards it to worker servers to execute under the master servers' control. The worker server sends the tasks to be performed to its master servers once the worker server completes its job, and the master servers may assign a new job according to the schedule. The experiment shows that MapReduce has a better performance in HFN than the tree-based hierarchical architecture.

3.4.6 CloudCube

As the basic structure of HFN is from BCube, a large number of switches are required in network establishment. To resolve this problem, CloudCube (Formu, 2009) was proposed based on HFN and BCube, which share the same structure as $\text{HFN}_{0(n,n)}$ if considering $\text{CloudCube}_{0(n)}$ as virtual servers. By contrast, CloudCube interchanges the positions of switches and servers in BCube to create a $\text{CloudCube}_{k(m,\text{CloudCube}_{0(n)})}$ architecture, where m denotes the number of switches connected by $\text{CloudCube}_{0(n)}$, and commonly, $m = n$. The number of potential servers in CloudCube is much larger than that in HFN, which effectively reduces the cost and enhances the scalability of DCN.

3.5 Optical/Wireless

Below are one optical and one wireless architecture for enhancing the DCN performance.

3.5.1 Optical switching DCN architecture (OSA)

Chen K *et al.* (2012) believed that if the network is able to dynamically change its topology and link bandwidth, then an unprecedented flexible architecture can be supported in DCN. Thereby, they introduced a novel optical switching architecture for DCN (called OSA) that uses the optical switching matrix (OSM), the wavelength selective switch (WSS), and wavelength division multiplexing (WDM). Fig. 15 shows the OSA architecture.

Most OSM modules are bipartite $N \times N$ matrices where any input port connects to any one

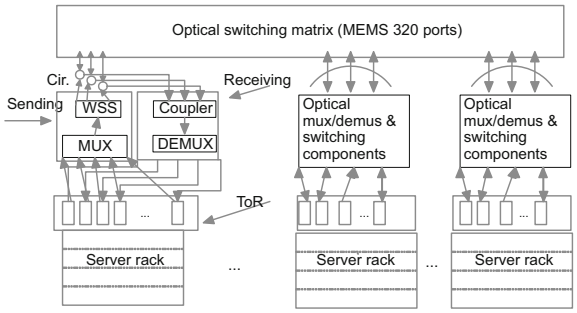

Fig. 15 A sample topology of OSA architecture

of the output ports. Nowadays, the micro-electro-mechanical switch (MEMS) has been widely used in OSM to reconfigure new input/output matching and connection within 10 ms by mechanically adjusting a microscopic array of mirrors. The WSS is a $1 \times N$ switch consisting of one common port and N wavelength ports to partition the set of wavelengths coming through the common port among the N wavelength ports.

OSA employs a shortest path routing scheme and hop-by-hop switching to ensure the network-wide connectivity in DCN. To reach remote indirectly connected ToRs, the first-hop ToR converts the forwarding data in fiber from optics to electronics signals for checking the packet header, and then towards data to next hop after converting the data packet from electronics back to optics signals. In addition, a central OSA manager is responsible for topology management, traffic and routing estimation and configurations.

Helios (Farrington *et al.*, 2011) and c-Through (Wang *et al.*, 2010) are well-known hybrid electrical-optical structures. In this hybrid model, each ToR connects to an electrical and an optical network at the same time. The electrical network is a two- or three-layer tree-based hierarchical structure with a certain oversubscription ratio. In the optical network, each ToR maintains a single optical connection to other ToRs, and this optical connection is of unrestricted capacity.

Optical switching has better potential performance than node switching in terms of data transmission speed, flexible topology, power-saving, and the bit ratio in long distance forwarding (Ikeda and Tsutsumi, 1995). Moreover, optical switching generates less heat to reduce the maintenance cost of cooling and radiating. Therefore, optical switching is an important research topic in DCN.

3.5.2 Wireless-DCN (WDCN)

Wireless technology can flexibly change network topology without re-cabling the layout; thereby, Ranachandran (2008) operated wireless technology in DCN. Later on, Kandula *et al.* (2009) described the Flyways architecture to de-congest and reduce data forwarding time between ToR switches in DCN. However, the separated wireless network has a hard job to meet all the requirements of DCN such as scalability, capacity, and fault-tolerance. For example, the bandwidth of a wireless network is commonly limited due to high traffic load and interference. Cui *et al.* (2011) proposed a hybrid Ethernet/wireless architecture in DCN, called WDCN.

To avoid excess antenna use and interference, Cui *et al.* (2011) considered each ToR as a wireless transmission unit (WTU) in WDCN (Fig. 16). Using 60 Hz wireless communication technology, Shin *et al.* (2012) proposed a fully wireless connection DCN, integrating switching fabric into server nodes to reduce the actual distance between ToRs and support fault-tolerance. They also replaced the network interface card (NIC) of a server to a Y-switch, and deployed these servers to circular structure racks. The above approaches can easily establish communication channels between the interior of racks, and create a mesh network structure. As this mesh network is a kind of Cayley graph (Alon and Roichman, 1994), it is also called the Cayley data center (CayleyDC).

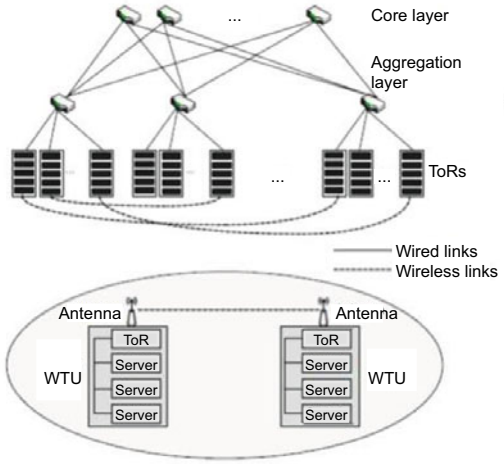

Fig. 16 A sample topology of WDCN architecture

Deploying wireless connection makes the network topology flexible and the cabling layout less complex. However, with a certain bandwidth, the

forwarding distance in a wireless network is limited and more overhead is generated due to broadcasting.

3.6 Randomly connected architecture

The randomly connected architecture is a random DCN where each switch or server can be connected randomly to some other switch.

3.6.1 Jellyfish

The architectures introduced (Al-Fares *et al.*, 2008; Niranjan Mysore *et al.*, 2009; Heller *et al.*, 2010) are all improvement of the tree-based hierarchical structure, and some common disadvantages exist. For example, network scale is restricted by the number of core routers, weakness in switch failure recovery, one-to-many and many-to-many communications, and cloud computing. The DCN architecture therefore tends to be a flat structure to modify the network structure from three layers to two layers or even one layer, e.g., mesh structure such as Jellyfish (Singla *et al.*, 2012).

Jellyfish constructs a random graph topology at the ToR switch layer, and each ToR switch i has k_i ports, of which r_i ports connect to other ToR switches and the remaining $k_i - r_i$ ports to servers. In the simplest case, each switch has the same number of ports and servers, i.e., $k = k_i$, $r = r_i$. When N is the number of ToR switches, a total of N_{kr} servers can be supported in DCN. Fig. 17 shows a topology of Jellyfish architecture.

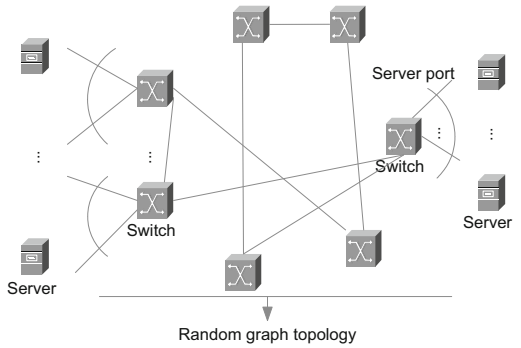

Fig. 17 A sample topology of Jellyfish architecture

Singla *et al.* (2012) pointed out that when the number of servers is less than 900, the number of servers Jellyfish supports is 27% more than that fat-tree supports, and the performance improves with the increase of the network scale. Jellyfish also has a shorter average path and larger bandwidth capacity than fat-tree, and a better performance in power saving. However, as Jellyfish is a random regular graph structure, the cabling layout issue is a big challenge, which limits the positions among the ToRs, and the implementation of optimal routing is also a challenge.

3.6.2 Scafida

Scafida (Gyarmati and Trinh, 2010) is another randomly connected DCN architecture inspired by scale-free networks. Scafida extends the original scale-free network generation algorithm of Barabási and Albert (1999) to meet the physical constraints of switches and routers that limit the node degrees in the network.

3.7 Analysis of DCN architectures

This subsection presents a qualitative analysis by comparing these architectures from six criteria: scale, bandwidth, fault tolerance, scalability, overhead, and cost of deployment. Scale, fault tolerance, and scalability are important design concerns as DCNs consist of a large amount of servers and network equipment and support a different range of cloud-based applications and services. Scale refers to the number of servers that are supported by the existing architecture. Fault-tolerance refers to whether the proposed architecture can effectively deal with the problems of the server, switch, and link failure. Scalability refers to whether the proposed architecture has a centralized node and whether it can easily deploy more devices. As a simple service in CDC is typically supported by the cooperation of multiple servers, bandwidth refers to a proportion of bandwidth between servers and network adapters that deeply affects the QoS. Overhead and cost of deployment are also important factors in DCN design as a reasonable cost according to the business requirements is considered essential for IT investment. In this subsection, overhead refers to the number of switches and links, and their cost. Cost of deployment refers to the workload of switch and server configuration, and the construction of basic equipment.

Table 2 compares the proposed DCN architectures from the above aspects. The detailed analysis is given as follows:

Table 2 A comparison of the proposed data center network (DCN) architectures

Architecture	Scale	Bandwidth	Fault-tolerance	Scalability	Overhead	Cost of deployment
Tree-based hierarchical	Small	Low	Bad	Bad	Very high	Very high
Fat-tree	Medium	Medium	Medium	Medium	High	High
ElasticTree	Medium	Medium	Medium	Medium	High	Very high
PortLand	Medium	Very high	Good	Medium	High	High
Jellyfish	Large	Very high	Good	Good	Medium	High
VL2	Large	Very high	Medium	Medium	High	High
Monsoon	Large	Very high	Medium	Medium	High	High
CamCube	Large	High	Good	Good	Very high	High
DCell	Large	High	Good	Good	High	High
FiConn	Large	High	Good	Good	Medium	High
BCube	Small	Very high	Very good	High	High	High
MDCube	Large	High	Good	Good	Very high	High
HFN	Small	Medium	Medium	Good	low	Medium
CloudCube	Small	Medium	Medium	Good	low	Medium
OSA	Small	Very high	Bad	Medium	High	Medium
WDCN	Small	Very high	Good	Medium	Medium	Medium

1. Network scale: Clos/Tree-based, VLB, and optical/wireless architectures connect servers and switches to establish a simple and easily connecting tree-based hierarchical topology. Servers play a full part only in data processing. In hierarchical recursive architectures, by contrast, each server is installed with one or more extra network adapters to establish a flexible, complicated, and specific network architecture. Servers not only process data but also respond to data transmission. In DCell, for example, servers are fully connected in identical layers, which makes it more scalable than tree-based architecture. The deployment, however, is a complicated mission for DCell because of the significant cabling layout. Note that the network scales of OSA and WDCN are smaller than those of others due to the higher deployment cost and limited wireless transmission range in optical/wireless architectures.

2. Bandwidth: The initial goal of the proposed architecture design is to deal with the bandwidth bottleneck in DCN. Therefore, VLB, hierarchical recursive, optical/wireless, and even the improved tree-based (such as fat-tree and Jellyfish) architectures, all have larger bandwidth than the original Clos/tree-based architectures. In addition, optical switches are advanced with ultra-high speed data forwarding, larger bandwidth, and lower loss compared with electrical switches.

3. Scalability: Clos/Tree-based architectures expand the scale of DCN by adding a number of ports and the number of levels on switches. They present the advantages of ease-of-wire but are limited by poor scalability and fault-tolerance. The improved tree-based architectures, such as fat-tree and VL2, solve the problems by increasing the number of switches in the aggregation layer, but the cabling layout becomes much more complex. By contrast, hierarchical recursive architectures have a limited scale of network, as the number of network adapters installed on a server is limited.

4. Overhead: Clos/Tree-based and VLB architectures use specialized routing equipment, such as the switch and router, to forward data, and the server is concerned only with data processing and storage. To effectively use lower level resources, those high level switches and routers are necessary to provide better data processing capacity and higher bandwidth. As servers participate in data forwarding in hierarchical recursive architectures, part of the CPU and memory resources of servers is consumed.

5. Cost of deployment: As hierarchical recursive architectures employ the server to transfer data, they have lower cost than tree-based architectures in the same cost/performance conditions. In addition, the cost of hierarchical recursive deployment may reduce with the development of the CPU and network adapter, such as integrating a module in the CPU to process a networking related task, or improving the autonomy process capacity of the network adapter to forward data without CPU. By contrast, in optical/wireless and tree-based architectures, a higher cost is generated by additional fiber optic or electro-optical transmitters/receiver, and highly advanced routers and switches are required to support higher

network performance in DCN.

Existing DCN architectures are all fixed (Wu *et al.*, 2012). They are advanced in one or more network evaluation metrics but may not be supported sufficiently in other metrics. It is still difficult to decide which architecture performs the best and whether it is suitable in a specific DCN.

4 Challenges and open research issues

With the existing data center network designs, we identify the key challenges and point out some open research issues that can be the subject of future research.

4.1 Key challenges

1. Congestion control: Cloud-oriented DCNs adopt TCP and Ethernet as their layer-4 and layer-2 transmission technologies. However, the broadcast nature of data transmissions in Ethernet causes significant traffic congestion, which makes the TCP retransmission mechanism unworkable, especially when a large number of packets are lost. TCP-Incast is a unique phenomenon observed in some cloud computing applications, such as MapReduce and cluster-based storage systems (Chen *et al.*, 2009). For instance, consider the example where multiple servers simultaneously communicate with a single client as a scenario. A large number of packets are dropped as the switch buffer overflows. Then the application throughput decreases rapidly due to the packet loss and TCP retransmission timeout (RTO). RTO may degrade application throughput by up to 90%.

2. Load balancing/Flow scheduling: The purpose of load balancing in cloud-oriented DCN is to distribute the workload to network equipment fairly, by routing traffic across multiple paths. As mentioned in Section 3, the cloud-oriented data center architectures, such as fat-tree (Al-Fares *et al.*, 2008) and Clos (Dally and Towles, 2004) network, often use dense multi-path topologies to provide large bandwidth for internal data exchange. In such networks, it is critical to employ effective load balancing schemes for fairly utilizing network resources.

In private or traditional data centers, workload patterns are relatively predictable. Typically, routing in such an environment is based on the shortest path algorithms, for example, open the shortest path first (OSPF). The shortest path from one node to another is calculated in advance without considering load balancing over multiple paths, and all the corresponding traffic is directed through this shortest path.

For cloud-oriented DCNs, several properties of cloud applications make the load balancing more highly complex than the traditional (Singh *et al.*, 2008). Prefix-routing is insufficient since workload patterns in cloud-oriented DCN are a priori unknown and variable for the network designer. Enterprises prefer to running their applications on commodity hardware, so the network can meet QoS without requiring software or protocol changes. Cloud computing providers use virtualization technology to efficiently multiplex customer's applications and processes across physical machines. It is difficult for customers to deal with inter-VM communication in a traditional application way.

4.2 Open research issues

1. Energy efficiency: As the number and the scale of data centers are growing explosively, energy conservation is emerging as an increasingly important global consensus issue. Based on a report submitted to Congress by the U.S. Environmental Protection Agency as part of the Energy Star program, networking devices in data centers in the United States accounted for 6.5 billion kW·h/year in 2012 (USEPA, 2012). How to build green and low consumption data centers has become a serious research issue. The consumers of energy in data centers include servers, networking equipment, power distribution, and cooling facilities. Most approaches (Beloglazov and Buyya, 2010; Chen Y *et al.*, 2012; Lee and Zomaya, 2012; Boru *et al.*, 2013) focus on making servers and cooling infrastructures more energy efficient. In contrast, the energy consumed by networking equipment is rarely considered, because networking equipment takes up a relatively small proportion of the data center's energy budget. As servers and cooling within data centers become more energy efficient, the percentage of data center power consumed by networking equipment is expected to grow.

2. Network optimizing: Bandwidth utilization and cabling complexity are becoming significant factors in the novel network architecture design. For example, the metric of bisection bandwidth has been widely used in DCN performance evaluation (Al-Fares *et al.*, 2008; Guo *et al.*, 2008; 2009; Katayama

et al., 2011), and the throughput in the aggregate layer measures the sum of the aggregated data flows when a network broadcast is conducted in tree-based DCN architectures. In traditional DCN, the cabling layout is simple. In cloud-oriented DCN, however, cabling is a critical issue because of the great number of nodes that have an impact on connecting efforts and maintenance. The issue of designing an optimized network structure for particular applications to increase its competitiveness in the era of cloud computing is yet to be addressed.

3. Novel network architectures: Network architecture in a distributed system has been studied extensively, and researchers have proposed a number of network structures (Frécon and Stenius, 1998; Foster *et al.*, 2002; Lian *et al.*, 2002; Tennenhouse and Wetherall, 2002). In DCN, the deployment of the existing mature network architectures needs to be analyzed and validated, especially those in the model of server-centric (Guo *et al.*, 2009). Novel network architectures for cloud-oriented DCN are also expected in further research.

4. Compatibility: In the actual deployment and upgrading of cloud-oriented DCN, purchasing devices with different capacities at different batch times is often considered for cost saving. Therefore, how to interconnect large-scale heterogeneous devices while ensuring the new DCN and existing networks cooperate efficiently is a major issue to be addressed.

5. Research and improvement of the DCN protocol: The management of the architecture of DCN is significantly different from the existing Internet architecture. The management of DCN is often accomplished in an instance. Thus, its global topology, data flow, failure, and various log information can be obtained to assist in protocol design and network architecture design. Novel protocols which are suitable for a specific DCN architecture can improve the efficiency of execution.

6. Automatic IP address assignment: Information about location and network topology in PortLand and BCube is stored at the server or switch, which improves the performance of routing. Therefore, traditional protocols such as the dynamic host configuration protocol (DHCP) (Droms, 1997) cannot be deployed in this condition. In addition, an automatic IP address assignment mechanism is required to reduce labor costs and the risk of configuration errors, since the manual configuration of such a large number of switches or servers is a time-consuming and tedious task. Therefore, proposing low-cost, high-reliability, and manageable automatic address configuration methods, regardless of known or unknown DCN architecture, is a challenging research perspective.

7. Future applications of optical switching and wireless transmission: The hybrid structure of optical/electrical switching is superior to traditional electrical switching architectures in terms of the cabling layout, design complexity, and energy consumption. However, optical equipment is still relatively expensive and is not yet deployed in DCN. Therefore, in addition to architecture design, reducing the cost is an important research perspective. Even though the architecture of a fully wireless layout has minimum complexity, designing a reliable and high-performance multi-hop network architecture is still a great challenge. In a hybrid architecture of wireless/wired, wireless technology can effectively alleviate the loading of hotspots, and efficient wireless routing of traffic demand is the challenging research perspective.

5 Conclusions

The data center network (DCN), as an important component of data centers, consists of a large number of hosted servers and switches connected with high-speed communication links. In recent years, the scale of DCN has constantly increased with the widespread use of cloud-based services and the unprecedented amount of data delivery in/between data centers, whereas traditional DCN architectures are ill-suited for cloud-oriented DCNs by lacking aggregate bandwidth and scalability, and are too costly for coping with the increasing demands of tenants in accessing the services.

In this paper we present a review of the recent research findings and technologies about DCN architectures for cloud computing. Motivated by a better support for data-intensive applications, how to optimize the interconnection of CDC becomes a fundamental issue. We describe the problems of the existing tree-based hierarchical architecture and challenges for cloud-oriented DCN, review the existing proposed architectures, and make a brief comparison from different aspects, including network scale, bandwidth, scalability, overhead, and cost of

deployment.

Although currently proposed architectures show some improvement in scalability, load balancing, and bandwidth capacity guarantees, they still face big challenges that are not solved yet. Important directions for future research in cloud-oriented DCNs include designing networks with high bandwidth capacity, scalability, and energy conservation, providing low cost, robustness, and strict guarantees of services, and implementing flexible management mechanisms to both providers and tenants.

References

Abu-Libdeh, H., Costa, P., Rowstron, A., *et al.*, 2010. Symbiotic routing in future data centers. *ACM SIGCOMM Comput. Commun. Rev.*, **40**(4):51-62. [doi:10.1145/1851275.1851191]

Al-Fares, M., Loukissas, A., Vahdat, A., 2008. A scalable, commodity data center network architecture. *ACM SIGCOMM Comput. Commun. Rev.*, **38**(4):63-74. [doi:10.1145/1402946.1402967]

Alon, N., Roichman, Y., 1994. Random Cayley graphs and expanders. *Random Struct. Algor.*, **5**(2):271-284. [doi:10.1002/rsa.3240050203]

Armbrust, M., Fox, A., Griffith, R., *et al.*, 2010. A view of cloud computing. *Commun. ACM*, **53**(4):50-58. [doi:10.1145/1721654.1721672]

Barabási, A.L., Albert, R., 1999. Emergence of scaling in random networks. *Science*, **286**(5439):509-512. [doi:10.1126/science.286.5439.509]

Beimborn, D., Miletzki, T., Wenzel, S., 2011. Platform as a service (PaaS). *Bus. Inform. Syst. Eng.*, **3**(6):381-384. [doi:10.1007/s12599-011-0183-3]

Beloglazov, A., Buyya, R., 2010. Energy efficient resource management in virtualized cloud data centers. Proc. 10th IEEE/ACM Int. Conf. on Cluster, Cloud and Grid Computing, p.826-831.

Bhardwaj, S., Jain, L., Jain, S., 2010. Cloud computing: a study of infrastructure as a service (IaaS). *Int. J. Eng. Inform. Technol.*, **2**(1):60-63.

Bilal, K., Khan, S.U., Kolodziej, J., *et al.*, 2012. A comparative study of data center network architectures. 26th European Conf. on Modelling and Simulation, p.526-532. [doi:10.7148/2012-0526-0532]

Bilal, K., Khan, S.U., Zhang, L., *et al.*, 2013a. Quantitative comparisons of the state-of-the-art data center architectures. *Concurr. Comput. Pract. Exp.*, **25**(12):1771-1783. [doi:10.1002/cpe.2963]

Bilal, K., Manzano, M., Khan, S.U., *et al.*, 2013b. On the characterization of the structural robustness of data center networks. *IEEE Trans. Cloud Comput.*, **1**(1):64-77.

Borthakur, D., 2007. The Hadoop Distributed File System: Architecture and Design. Available from http://svn.eu.apache.org [Accessed on Jan. 13, 2014].

Boru, D., Kliazovich, D., Granelli, F., *et al.*, 2013. Energy-efficient data replication in cloud computing datacenters. IEEE Globecom Int. Workshop on Cloud Computing Systems, Networks, and Applications, p.446-451.

Buxmann, P., Hess, T., Lehmann, S., 2008. Software as a service. *Wirtschaftsinformatik*, **50**(6):500-503. [doi:10.1007/s11576-008-0095-0]

Buyya, R., Yeo, C.S., Venugopal, S., 2008. Market-oriented cloud computing: vision, hype, and reality for delivering IT services as computing utilities. 10th IEEE Int. Conf. on High Performance Computing and Communications, p.5-13.

Chang, F., Dean, J., Ghemawat, S., *et al.*, 2008. Bigtable: a distributed storage system for structured data. *ACM Trans. Comput. Syst.*, **26**(2):1-26. [doi:10.1145/1365815.1365816]

Chen, K., Singla, A., Singh, A., *et al.*, 2012a. OSA: an optical switching architecture for data center networks with unprecedented flexibility. Proc. 9th USENIX Conf. on Networked Systems Design and Implementation.

Chen, Y., Griffith, R., Liu, J., *et al.*, 2009. Understanding TCP incast throughput collapse in datacenter networks. Proc. 1st ACM Workshop on Research on Enterprise Networking, p.73-82. [doi:10.1145/1592681.1592693]

Chen, Y., Alspaugh, S., Borthakur, D., *et al.*, 2012. Energy efficiency for large-scale MapReduce workloads with significant interactive analysis. Proc. 7th ACM European Conf. on Computer Systems, p.43-56. [doi:10.1145/2168836.2168842]

Cisco Data Center, 2007. Infrastructure 2.5 Design Guide.

Clos, C., 1953. A study of non-blocking switching networks. *Bell Syst. Techn. J.*, **32**(2):406-424. [doi:10.1002/j.1538-7305.1953.tb01433.x]

Cui, Y., Wang, H., Cheng, X., *et al.*, 2011. Wireless data center networking. *IEEE Wirel. Commun.*, **18**(6):46-53. [doi:10.1109/MWC.2011.6108333]

Dally, W.J., Towles, B., 2004. Principles and Practices of Interconnection Networks. Morgan Kaufmann, San Francisco, CA, USA.

Dean, J., Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM*, **51**(1):107-113. [doi:10.1145/1327452.1327492]

Ding, Z., Guo, D., Liu, X., *et al.*, 2012. A MapReduce-supported network structure for data centers. *Concurr. Comput. Pract. Exp.*, **24**(12):1271-1295. [doi:10.1002/cpe.1791]

Droms, R., 1997. Dynamic Host Configuration Protocol. RFC Editor, United States.

Farrington, N., Porter, G., Radhakrishnan, S., *et al.*, 2011. Helios: a hybrid electrical/optical switch architecture for modular data centers. *ACM SIGCOMM Comput. Commun. Rev.*, **41**(4):339-350.

Formu, J., 2009. Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration.

Foster, I., Kesselman, C., Nick, J., *et al.*, 2002. Grid services for distributed system integration. *Computer*, **35**(6):37-46. [doi:10.1109/MC.2002.1009167]

Frécon, E., Stenius, M., 1998. Dive: a scaleable network architecture for distributed virtual environments. *Distr. Syst. Eng.*, **5**(3):91-100. [doi:10.1088/0967-1846/5/3/002]

Gantz, J., Reinsel, D., 2012. The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. IDC iView: IDC Analyze the Future.

Ghemawat, S., Gobioff, H., Leung, S.T., 2003. The Google File System. *ACM SIGOPS Oper. Syst. Rev.*, **37**(5):29-43. [doi:10.1145/1165389.945450]

Greenberg, A., Hamilton, J., Maltz, D.A., *et al.*, 2008a. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM Comput. Commun. Rev.*, **39**(1):68-73. [doi:10.1145/1496091.1496103]

Greenberg, A., Lahiri, P., Maltz, D., *et al.*, 2008b. Towards a next generation data center architecture: scalability and commoditization. Proc. ACM Workshop on Programmable Routers for Extensible Services of Tomorrow, p.57-62. [doi:10.1145/1397718.1397732]

Greenberg, A., Hamilton, J.R., Jain, N., *et al.*, 2009. Vl2: a scalable and flexible data center network. *ACM SIGCOMM Comput. Commun. Rev.*, **39**(4):51-62. [doi:10.1145/1594977.1592576]

Guo, C., Wu, H., Tan, K., *et al.*, 2008. DCell: a scalable and fault-tolerant network structure for data centers. *ACM SIGCOMM Comput. Commun. Rev.*, **38**(4):75-86. [doi:10.1145/1402946.1402968]

Guo, C., Lu, G., Li, D., *et al.*, 2009. BCube: a high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Comput. Commun. Rev.*, **39**(4):63-74. [doi:10.1145/1594977.1592577]

Gyarmati, L., Trinh, T., 2010. Scafida: a scale-free network inspired data center architecture. *ACM SIGCOMM Comput. Commun. Rev.*, **40**(5):4-12. [doi:10.1145/1880153.1880155]

Heller, B., Seetharaman, S., Mahadevan, P., *et al.*, 2010. ElasticTree: saving energy in data center networks. Proc. 7th USENIX Conf. on Networked Systems Design and Implementation, p.19-21.

Ikeda, T., Tsutsumi, O., 1995. Optical switching and image storage by means of azobenzene liquid-crystal films. *Science*, **268**(5219):1873-1875. [doi:10.1126/science.268.5219.1873]

Isard, M., Budiu, M., Yu, Y., *et al.*, 2007. Dryad: distributed data-parallel programs from sequential building blocks. *ACM SIGOPS Operat. Syst. Rev.*, **41**(3):59-72. [doi:10.1145/1272998.1273005]

Jericho Forum, 2009. Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration.

Kandula, S., Padhye, J., Bahl, P., 2009. Flyways to Decongest Data Center Networks.

Katayama, Y., Takano, K., Kohda, Y., *et al.*, 2011. Wireless data center networking with steered-beam mm wave links. IEEE Wireless Communications and Networking Conf., p.2179-2184.

Lee, Y.C., Zomaya, A.Y., 2012. Energy efficient utilization of resources in cloud computing systems. *J. Supercomput.*, **60**(2):268-280. [doi:10.1007/s11227-010-0421-3]

Li, D., Guo, C., Wu, H., *et al.*, 2009. Ficonn: using backup port for server interconnection in data centers. IEEE INFOCOM, p.2276-2285.

Li, W., Svard, P., 2010. REST-based SOA application in the cloud: a text correction service case study. World Congress on Services, p.84-90.

Lian, F.L., Moyne, J., Tilbury, D., 2002. Network design consideration for distributed control systems. *IEEE Trans. Contr. Syst. Technol.*, **10**(2):297-307. [doi:10.1109/87.987076]

Manzano, M., Bilal, K., Calle, E., *et al.*, 2013. On the connectivity of data center networks. *IEEE Commun. Lett.*, **17**(11):2172-2175. [doi:10.1109/LCOMM.2013.091913.131176]

Niranjan Mysore, R., Pamboris, A., Farrington, N., *et al.*, 2009. Portland: a scalable fault-tolerant layer 2 data center network fabric. *ACM SIGCOMM Comput. Commun. Rev.*, **39**(4):39-50. [doi:10.1145/1594977.1592575]

Popa, L., Ratnasamy, S., Iannaccone, G., *et al.*, 2010. A cost comparison of datacenter network architectures. Proc. 6th Int. Conf. Co-NEXT, Article 16. [doi:10.1145/1921168.1921189]

Ranachandran, K., 2008. 60 GHz Data-Center Networking: Wireless=>Worryless. Technical Report, NEC Laboratories America, Inc.

Redkar, T., Guidici, T., 2011. Windows Azure Platform. Apress.

Rimal, B., Choi, E., Lumb, I., 2009. A taxonomy and survey of cloud computing systems. 5th Int. Joint Conf. on INC, IMS and IDC, p.44-51. [doi:10.1109/NCM.2009.218]

Shin, J.Y., Sirer, E.G., Weatherspoon, H., *et al.*, 2012. On the feasibility of completely wireless datacenters. Proc. 8th ACM/IEEE Symp. on Architectures for Networking and Communications Systems, p.3-14. [doi:10.1145/2396556.2396560]

Singh, A., Korupolu, M., Mohapatra, D., 2008. Server-storage virtualization: integration and load balancing in data centers. Proc. ACM/IEEE Conf. on Supercomputing, p.53.

Singla, A., Hong, C.Y., Popa, L., *et al.*, 2012. Jellyfish: networking data centers randomly. Proc. 9th USENIX Conf. on Networked Systems Design and Implementation, p.17.

Tarantino, A., 2012. Point-of-view paper: high tech's innovative approach to sustainability. *Int. J. Innov. Sci.*, **4**(1):37-40. [doi:10.1260/1757-2223.4.1.37]

Tennenhouse, D., Wetherall, D., 2002. Towards an active network architecture. Proc. DARPA Active Networks Conf. and Exposition, p.2-15. [doi:10.1109/DANCE.2002.1003480]

Tschudi, W., Xu, T., Sartor, D., *et al.*, 2004. Energy Efficient Data Centers. Lawrence Berkeley National Laboratory.

Tziritas, N., Xu, C.Z., Loukopoulos, T., *et al.*, 2013. Application-aware workload consolidation to minimize both energy consumption and network load in cloud environments. 42nd IEEE Int. Conf. on Parallel Processing, p.449-457.

USEPA, 2012. 2012 Annual Report—US Environmental Protection Agency.

Vahdat, A., Al-Fares, M., Farrington, N., *et al.*, 2010. Scale-out networking in the data center. *IEEE Micro*, **30**(4):29-41. [doi:10.1109/MM.2010.72]

Valiant, L.G., 1990. A bridging model for parallel computation. *Commun. ACM*, **33**(8):103-111. [doi:10.1145/79173.79181]

Wang, G., Andersen, D.G., Kaminsky, M., *et al.*, 2010. C-through: part-time optics in data centers. *ACM SIGCOMM Comput. Commun. Rev.*, **40**(4):327-338. [doi:10.1145/1851275.1851222]

Wu, H., Lu, G., Li, D., *et al.*, 2009. MDCube: a high performance network structure for modular data center interconnection. Proc. 5th Int. Conf. on Emerging Networking Experiments and Technologies, p.25-36. [doi:10.1145/1658939.1658943]

Wu, K., Xiao, J., Ni, L.M., 2012. Rethinking the architecture design of data center networks. *Front. Comput. Sci.*, **6**(5):596-603.

Zahariev, A., 2009. Google APP Engine. Helsinki University of Technology, Helsinki, Finland.